



Utilizando LLM integrado ao framework Scrum na gestão de projetos ágeis

Breno Teixeira Conceição Souza ^{*}, Caio Bispo dos Reis [†], Daniel
de Jesus Machado [‡], Sérgio Luiz Teixeira Nunes Júnior [§]

Análise e Desenvolvimento de Sistemas
Escola de Tecnologias
Universidade Católica do Salvador (UCSAL)
Av. Prof. Pinto de Aguiar, 2589 Pituvaçu, CEP: 41740-090
Salvador/BA, Brasil

2025

^{*}breno.souza@ucsal.edu.br

[†]caiobispo.reis@ucsal.edu.br

[‡]danieljesus.machado@ucsal.edu.br

[§]sergioluiz.junior@ucsal.edu.br

Resumo

Este trabalho apresenta o desenvolvimento de uma ferramenta baseada em Modelos de Linguagem de Grande Escala (LLM) para otimizar a gestão de projetos com o framework Scrum. A metodologia adotada foi a pesquisa aplicada tecnológica, consistindo no desenvolvimento de uma aplicação web utilizando Next.js, Node.js e a API da Groq, integrada a um quadro Kanban interativo. Diante de desafios como a sobrecarga operacional de líderes ágeis, a ferramenta propõe a automação da criação de backlogs e estimativas. Os resultados qualitativos sugerem que a solução tem potencial para reduzir o esforço manual na estruturação de projetos, embora a supervisão humana permaneça indispensável para validar as sugestões da IA. A pesquisa contribui para demonstrar a viabilidade prática da integração entre IA Generativa e metodologias ágeis.

Palavras-chave: Gestão de Projetos Ágeis. Scrum. Inteligência Artificial. LLM. Engenharia de Software.

Sumário

Sumário	3	
1	Introdução	4
1.1	Problematização	4
1.2	Objetivos	5
1.2.1	Objetivo geral	5
1.2.2	Objetivos específicos	5
1.3	Metodologia	5
1.3.1	Levantamento bibliográfico	6
1.3.2	Desenvolvimento da ferramenta	6
1.3.3	Análise de resultados	6
2	Fundamentação teórica	6
2.1	Gestão de projetos ágeis	7
2.2	Scrum	7
2.3	Modelos de Linguagem de Grande Escala (LLM)	9
2.4	Scrum com Inteligência Artificial / LLMs	10
2.5	Considerações sobre o capítulo	11
3	A ferramenta	11
3.1	Arquitetura e tecnologias utilizadas	11
3.2	Desenvolvimento do front-end e usabilidade	14
3.3	Integração com Inteligência Artificial	14
3.4	Estrutura de classes e serviços	15
3.5	Comparativo com ferramentas de Scrum existentes	15
3.6	Resultados obtidos	16
3.7	Avaliação da ferramenta	16
4	Trabalhos relacionados	17
4.1	Revisão dos estudos selecionados	17
4.2	Análise comparativa de abordagens	18
4.3	Relação com o trabalho desenvolvido	18
5	Considerações finais	19
5.1	Resultados obtidos	19
5.2	Limitações do estudo	20
5.3	Sugestões para trabalhos futuros	20
	REFERÊNCIAS	22

1 Introdução

Com o crescimento emergente do uso de Inteligência Artificial em diversas áreas da vida, sua origem muitas vezes é esquecida. As primeiras histórias e mitos de “seres artificiais” se passam na Grécia Antiga, em que na mitologia grega Hefesto, o deus do fogo e da metalurgia era retratado como criador de autômatos de metal para ajudá-lo em suas tarefas (MAYOR, 2018). Transitando do mito para a ciência, o primeiro registro formal relacionado à IA foi em 1943, com Walter Pitts e Warren McCulloch contribuindo para o conceito de neurônios artificiais – componentes computacionais inspirados no sistema nervoso humano – que serviu como base para a criação das redes neurais, e posteriormente, para grande parte do conceito da Inteligência Artificial (MCCULLOCH; PITTS, 1943).

Em 1950, o matemático inglês Alan Turing, mundialmente conhecido como o pai da computação, publicou o artigo *Computing Machinery and Intelligence (Máquinas Computacionais e Inteligência)*, no qual era proposto o *Teste de Turing*, também conhecido como “Jogo da Imitação”, no qual uma máquina era testada em relação à capacidade de executar um comportamento próximo de um ser humano (TURING, 1950).

No contexto do avanço tecnológico, a organização e planejamento de projetos é crucial para se garantir a eficiência do processo, e com evolução das formas de se trabalhar em equipe e particionamento de tarefas, surgiu o Scrum, uma estruturação ágil de gestão de projetos que é estruturado em ciclos curtos de trabalho chamados *Sprints*. Contudo, mesmo com a utilização desse método, ainda surgem dificuldades no processo de construção do projeto, como resistência a mudanças (PRASETYO et al., 2024), falta de envolvimento do proprietário do produto, comunicação ineficiente, dependência de outros profissionais (GARCÍA; LEE, 2023), dentre outros.

O termo Scrum foi utilizado pela primeira vez no ano de 1986, no artigo *The New New Product Development Game (O Novo Jogo de Desenvolvimento de Novos Produtos)* publicado pela revista *The Harvard Business Review* – uma revista que mostra insights e reflexões sobre as melhores práticas na gestão de negócios – pelos autores Hirotaka Takeuchi e Ikujiro Nonaka (TAKEUCHI; NONAKA, 1986). Este artigo fazia a comparação de equipes de alto desempenho com uma jogada existente no esporte Rugby de mesmo nome: Scrum.

Anos depois, em 1995, Ken Schwaber (desenvolvedor de software, gerente de produto e consultor) e Jeff Sutherland (formações diversas, na área de tecnologia, militar e medicina) publicaram o *Scrum Development Process*, um livro contendo todos os princípios dessa metodologia (SCHWABER; SUTHERLAND, 1995).

1.1 Problematização

Com o passar dos anos e evolução tanto humana como tecnológica, a necessidade de inserir a Inteligência Artificial também na área de gestão de projetos surge com as de-

mandas por mais agilidade nas construções de projetos – mais especificamente nesse caso, na área de metodologias ágeis, para automatizar funções que podem ser realizadas por agentes autônomos, e focar nas tarefas que requerem a participação humana (ALLIATA; ARCURI; BELLER, 2025). Conseqüentemente com esses progressos, cada vez mais irá se tornar uma ferramenta de trabalho não mais opcional, mas sim imprescindível, e que acompanhe os passos largos da evolução humana.

Apesar da grande adoção do Scrum, as equipes de desenvolvimento ainda enfrentam dificuldades causadas pela dependência de processos manuais. Atividades como detalhamento de tarefas, classificação técnica e especificação de prazos acabam consumindo um tempo significativo (SCHRÖDER, 2023), principalmente dos líderes dos projetos (Scrum Master e Product Owner), desviando a atenção de questões mais prioritárias (ALLIATA; ARCURI; BELLER, 2025).

A principal problemática está na dificuldade em manter a agilidade para atender a todas as atividades, sem comprometer a eficiência e qualidade do processo (PRASETYO et al., 2024; KARNOUSKOS, 2024). Partindo desse ponto, surge o questionamento: "Como os Modelos de Linguagem de Grande Escala (LLM) podem ser utilizados para aprimorar o uso do Scrum, reduzindo assim a sobrecarga da equipe?".

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver e avaliar uma ferramenta baseada em um Modelo de Linguagem de Grande Escala (LLM) para otimizar a eficácia da gestão de projetos que utilizam o framework Scrum.

1.2.2 Objetivos específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar um levantamento bibliográfico sobre Modelos de Linguagem de Grande Escala (LLM) e os desafios na adoção do Scrum;
- Implementar um agente de Inteligência Artificial capaz de auxiliar em tarefas operacionais do Scrum;
- Avaliar a aplicabilidade e os resultados obtidos com o uso da ferramenta desenvolvida no contexto de gestão de projetos.

1.3 Metodologia

A metodologia adotada para o desenvolvimento deste trabalho classifica-se como pesquisa aplicada, de natureza exploratória e tecnológica. O processo foi estruturado em

três etapas principais para garantir a fundamentação teórica e a viabilidade prática da solução proposta: levantamento bibliográfico, desenvolvimento da ferramenta, e análise de resultados.

1.3.1 Levantamento bibliográfico

Foram realizadas revisões bibliográficas sistemáticas para compreender o estado da arte das tecnologias envolvidas. Foram consultadas bases de dados acadêmicas como Google Scholar, IEEE Xplore e ResearchGate, buscando trabalhos que relacionassem o framework Scrum, Gestão de Projetos Ágeis e Modelos de Linguagem de Grande Escala (LLM). Esta etapa serviu para identificar as principais dificuldades na adoção do Scrum e como a Inteligência Artificial pode mitigar esses problemas.

1.3.2 Desenvolvimento da ferramenta

A etapa seguinte consiste no desenvolvimento do agente de IA. A solução foi projetada para integrar-se ao fluxo de trabalho do Scrum, utilizando APIs de LLMs para processar linguagem natural, e automatizar tarefas. O desenvolvimento seguiu práticas ágeis, com iterações focadas na construção das funcionalidades de organização de tarefas, priorização de backlog e classificação de itens (Front-end/Back-end).

1.3.3 Análise de resultados

Por fim, a metodologia incluiu a validação da ferramenta desenvolvida. A avaliação foi realizada por meio de testes funcionais e qualitativos conduzidos pela própria equipe de desenvolvimento. Foram simulados cenários de criação de projetos de software (ex: e-commerce, sistema de gestão), analisando se a IA gerava tarefas coerentes com o escopo solicitado e se a interface gráfica respondia adequadamente às ações de priorização e movimentação de cards.

2 Fundamentação teórica

Este capítulo estabelece a base conceitual do projeto, detalhando as tecnologias e metodologias que se interconectam para formar a solução proposta. Os conceitos de Inteligência Artificial, LLM, Scrum e Gestão de Projetos Ágeis são apresentados para contextualizar a integração de um agente inteligente em um fluxo de trabalho ágil, para otimizar os fluxos e reduzir tempo de trabalho, fazendo com que os *stakeholders* foquem nas partes que necessitam da “mão humana”.

2.1 Gestão de projetos ágeis

Tradicionalmente, a abordagem para a materialização de um projeto consistia em planejar cada etapa minuciosamente. A equipe recebia um plano completo e sua missão era executá-lo fielmente, com pouca margem para alterações durante o percurso. O Manifesto Ágil, proposto por Beck et al. (2001), marcou a transição para uma abordagem mais flexível, ao introduzir princípios que valorizam adaptação contínua e foco na entrega de valor.

A principal mudança de paradigma do ágil é a priorização da colaboração com o cliente e a capacidade de resposta eficiente a mudanças, em detrimento do seguimento rígido de um escopo inicial. Highsmith (2002) destaca que a essência da agilidade é justamente a habilidade de responder rapidamente a mudanças, reconhecendo que, em projetos complexos, é impossível prever todas as variáveis antecipadamente. Assim, a adaptação torna-se um fator crucial para o sucesso.

Na prática, isso se traduz em trabalhar com ciclos curtos de planejamento e execução, onde o objetivo é entregar partes funcionais do produto de forma incremental. Esse ciclo constante de construção, feedback e ajuste é o motor do ágil, pois permite que a equipe aprenda e melhore continuamente, mitigando riscos e garantindo que o resultado final realmente atenda às necessidades do usuário (BECK et al., 2001).

Além disso, a popularização do ágil no mercado é evidente. Segundo o relatório State of Agile (??), mais de 90% das empresas utilizam práticas ágeis em algum nível, demonstrando a consolidação da metodologia em diferentes setores. Para organizar essa dinâmica, existem diversas estruturas, como o Kanban, que utiliza um método visual em colunas como "a fazer", "pendente", "feito", e cartões representando tarefas, para otimizar o fluxo de trabalho (Atlassian, 2025).

2.2 Scrum

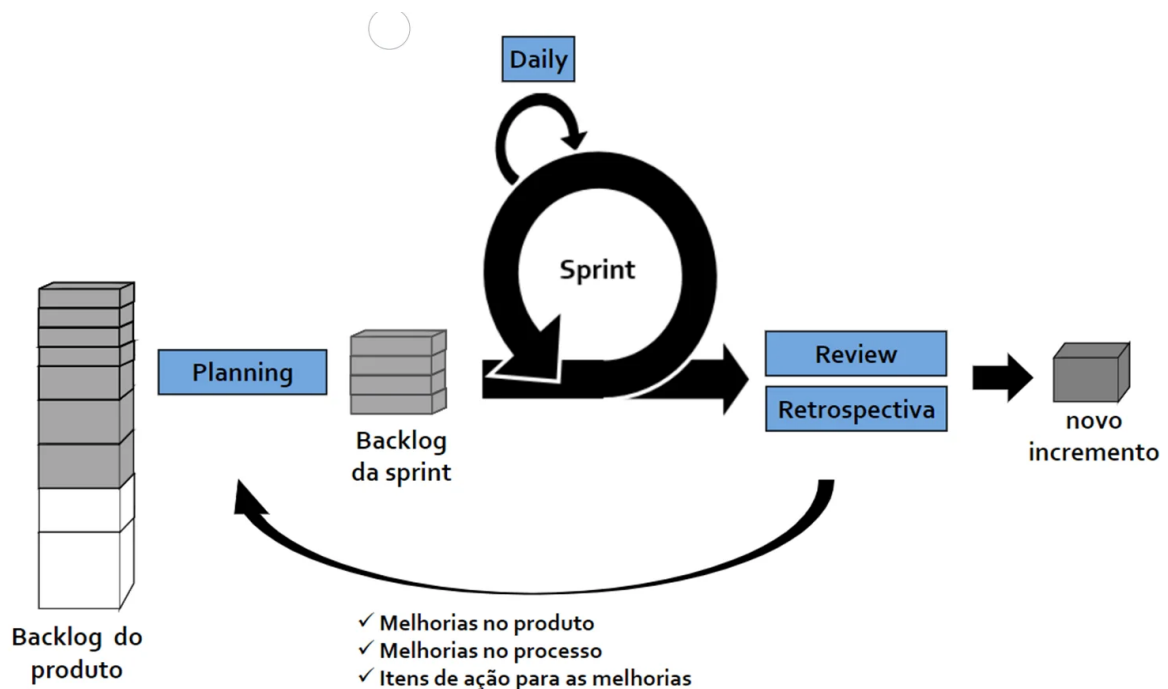
O Scrum é um framework ágil de gerenciamento de projetos desenvolvido por Ken Schwaber e Jeff Sutherland na década de 1990, projetado para lidar com a complexidade e a imprevisibilidade inerentes ao desenvolvimento de software (SCHWABER; SUTHERLAND, 2020). Diferentemente de metodologias tradicionais, que seguem um planejamento rígido e sequencial, o Scrum propõe uma abordagem iterativa e incremental, estruturada em ciclos curtos de trabalho denominados Sprints, nos quais equipes multifuncionais colaboram para entregar incrementos de produto potencialmente utilizáveis (SCHWABER; SUTHERLAND, 2020).

No centro do framework estão os papéis: o *Product Owner* (PO), responsável por definir e priorizar o backlog do produto; o *Scrum Master* (SM), que atua como facilitador e garante a aplicação correta do framework; e o Time de Desenvolvimento, que

transforma os requisitos priorizados em incrementos de produto utilizáveis (SCHWABER; SUTHERLAND, 2020).

Nos Eventos, englobam-se o *Sprint Planning* (Planejamento), no qual se define o que será desenvolvido; o *Daily Scrum* (Daily), reunião diária para alinhamento da equipe; a *Sprint Review* (Review), voltada à apresentação dos resultados aos *stakeholders*; e a *Sprint Retrospective* (Retrospectiva), destinada à identificação de melhorias de processo. Os artefatos do Scrum asseguram a transparência do trabalho. Entre eles, destacam-se: o *product backlog* (Backlog do Produto), que contém a lista priorizada de funcionalidades; o *sprint backlog* (Backlog da Sprint), com os itens selecionados para a sprint em andamento; e o incremento, que representa a soma de todos os itens concluídos (COHN, 2010). O ciclo de funcionamento desses eventos pode ser visualizado de forma representada na figura 1.

Figura 1 – Fluxo de Funcionamento do Framework Scrum



Fonte: Adaptado de Ribas (2025).

O Scrum fundamenta-se em três pilares: transparência, inspeção e adaptação que sustentam a dinâmica do framework e garantem que a equipe possa responder rapidamente a mudanças de escopo, requisitos ou prioridades (SCHWABER; SUTHERLAND, 2020). Isso faz do Scrum uma abordagem especialmente eficaz em cenários de incerteza e complexidade, típicos da engenharia de software contemporânea.

Apesar das vantagens, sua aplicação prática ainda enfrenta desafios, como resistência a mudanças organizacionais, baixa participação dos *stakeholders*, problemas de comunicação entre equipes e dependência de profissionais externos para tomada de decisão (COHN, 2010). Esses fatores podem comprometer a eficiência do framework, reforçando

a necessidade de mecanismos que aprimorem a comunicação, a clareza de requisitos e o acompanhamento dos impedimentos. Nesse contexto, a integração de LLMs ao Scrum surge como uma alternativa promissora.

2.3 Modelos de Linguagem de Grande Escala (LLM)

A Inteligência Artificial (IA) é um campo da ciência da computação que busca desenvolver sistemas capazes de executar tarefas que tradicionalmente requerem inteligência humana, como reconhecimento de padrões, tomada de decisão e processamento de linguagem natural. Dentro desse campo, o Aprendizado de Máquina (*Machine Learning* – ML) destaca-se como uma abordagem que permite que algoritmos aprendam a partir de dados, identificando padrões e realizando previsões sem necessidade de programação explícita para cada caso (BOMMASANI et al., 2021).

Os Modelos de Linguagem de Grande Escala (*Large Language Models* – LLMs) representam um dos principais avanços recentes no campo da Inteligência Artificial e do Processamento de Linguagem Natural (PLN). Esses modelos são baseados em arquiteturas de redes neurais profundas, em especial nos *transformers*, introduzidos por Vaswani et al. (2017), que revolucionaram a forma de lidar com dados textuais ao possibilitar a captura de relações complexas entre palavras em diferentes contextos (BOMMASANI et al., 2021).

A evolução da IA tem sido impulsionada pelo avanço computacional e pela disponibilidade massiva de dados, permitindo o surgimento de modelos mais complexos e precisos. Essa evolução abriu espaço para aplicações em diversas áreas, incluindo saúde, educação, finanças, e mais recentemente, gestão de projetos (BOMMASANI et al., 2021; KARNOUSKOS, 2024).

Exemplos notáveis de LLMs incluem o GPT (*Generative Pre-trained Transformer*), desenvolvido pela OpenAI, cuja primeira versão foi apresentada por Radford et al. (2018); o BERT (*Bidirectional Encoder Representations from Transformers*), criado pelo Google e introduzido por Devlin et al. (2019); e o LLaMA (*Large Language Model Meta AI*), desenvolvido pela Meta, lançado por Touvron et al. (2023).

Cada um desses modelos possui características próprias: enquanto o GPT (RADFORD et al., 2018) se destaca na geração de texto coerente em múltiplos contextos, o BERT (DEVLIN et al., 2019) foi projetado para compreensão bidirecional da linguagem, tornando-se referência em tarefas de classificação e análise semântica, e o LLaMA (TOUVRON et al., 2023) busca maior eficiência de treinamento e adaptação em cenários específicos.

De acordo com Bommasani et al. (2021), os LLMs são treinados com enormes volumes de dados textuais, muitas vezes coletados de livros, artigos, sites e outros repositórios digitais, permitindo que adquiram uma capacidade emergente de generalização. Isso significa que, além de tarefas específicas para as quais foram treinados, esses modelos conseguem

desempenhar novas funções sem necessidade de reprogramação ou treinamento adicional extenso, fenômeno conhecido como *few-shot learning* ou *zero-shot learning* (BOMMASANI et al., 2021).

Entre as principais aplicações dos LLMs estão:

- Geração automática de texto coerente e contextualizado;
- Criação de resumos de documentos extensos;
- Tradução automática de idiomas;
- Apoio na análise semântica de grandes volumes de dados;
- Assistência em tomada de decisão baseada em linguagem.

Segundo Bender et al. (2021), esses modelos não "compreendem" a linguagem no sentido humano, mas aprendem padrões estatísticos que permitem prever a próxima palavra ou estrutura mais provável em um texto. Apesar disso, na prática, eles demonstram uma utilidade significativa em diversos setores, como educação, finanças e gestão de projetos.

No entanto, é fundamental destacar também as limitações dos LLMs. Eles podem reproduzir vieses presentes nos dados de treinamento (CALISKAN; BRYSON; NARAYANAN, 2017), gerar informações incorretas ou não verificáveis (alucinações), e ainda apresentam desafios em termos de consumo energético e sustentabilidade, devido à alta demanda computacional para treinamento (STRUBELL; GANESH; MCCALLUM, 2019).

Dessa forma, a aplicação de LLMs em contextos organizacionais, como no framework Scrum, exige supervisão humana, diretrizes éticas e metodologias de validação contínua para garantir resultados confiáveis e úteis (ALLIATA; ARCURI; BELLER, 2025). Assim, os LLMs oferecem não apenas recursos de automação, mas também capacidade de análise e suporte à tomada de decisão, características que tornam sua aplicação promissora no contexto do Scrum, onde a comunicação, a clareza de requisitos e a adaptação rápida são fatores críticos para o sucesso do projeto (BENDER et al., 2021).

2.4 Scrum com Inteligência Artificial / LLMs

Embora o Scrum apresente vantagens significativas em relação a métodos tradicionais, sua aplicação em projetos de larga escala ainda envolve desafios, como a coordenação entre múltiplas equipes, a manutenção da comunicação eficiente e o alinhamento de metas organizacionais (SCHWABER; SUTHERLAND, 2020). Nesse contexto, a integração de tecnologias emergentes, como os Modelos de Linguagem de Grande Escala (LLMs), surge como um recurso promissor.

Segundo Cinkusz, Chudziak e Niewiadomska-Szynkiewicz (2025), os LLMs têm remodelado a engenharia de software ao introduzir capacidades cognitivas que auxiliam

no raciocínio, no planejamento e no suporte ao trabalho em equipe. A combinação entre agentes cognitivos e modelos de linguagem possibilita um ambiente mais inteligente de apoio à gestão ágil, promovendo maior eficiência na organização dos requisitos, na comunicação entre equipes e na adaptação contínua às mudanças do projeto.

Essa integração pode fornecer uma visão mais aprofundada sobre os fluxos de trabalho, auxiliando não apenas na produtividade e qualidade das entregas, mas também na agilidade com que equipes Scrum respondem a objetivos em evolução. Assim, a aplicação de LLMs no contexto do Scrum configura-se como um campo emergente de pesquisa, com potencial para transformar práticas consolidadas de gestão ágil em processos ainda mais responsivos e inteligentes (CINKUSZ; CHUDZIAK; NIEWIADOMSKA-SZYNKIEWICZ, 2025; ALLIATA; ARCURI; BELLER, 2025).

2.5 Considerações sobre o capítulo

Neste capítulo, foram estabelecidos os conceitos e fundamentos deste trabalho. Foi evidenciada a evolução da Inteligência Artificial até o modelo de Linguagem de Grande Escala (LLM), seus recursos, formas de aprendizado e exemplos de LLM; a relevância do Scrum, seus diferenciais, papéis, etapas, vantagens e desafios; a junção entre LLM e Scrum; e mostrada a importância da Gestão de Projetos Ágeis, o que mudou com a sua chegada ao mercado, e seus diferenciais. Tais conceitos serão mostrados na prática no código desenvolvido neste projeto.

3 A ferramenta

Este capítulo aborda as tecnologias utilizadas na ferramenta desenvolvida, sua estruturação, as funcionalidades que a IA desempenha, os comparativos com outras ferramentas existentes de Scrum, os resultados obtidos e avaliação, a partir do desenvolvimento e utilização da mesma.

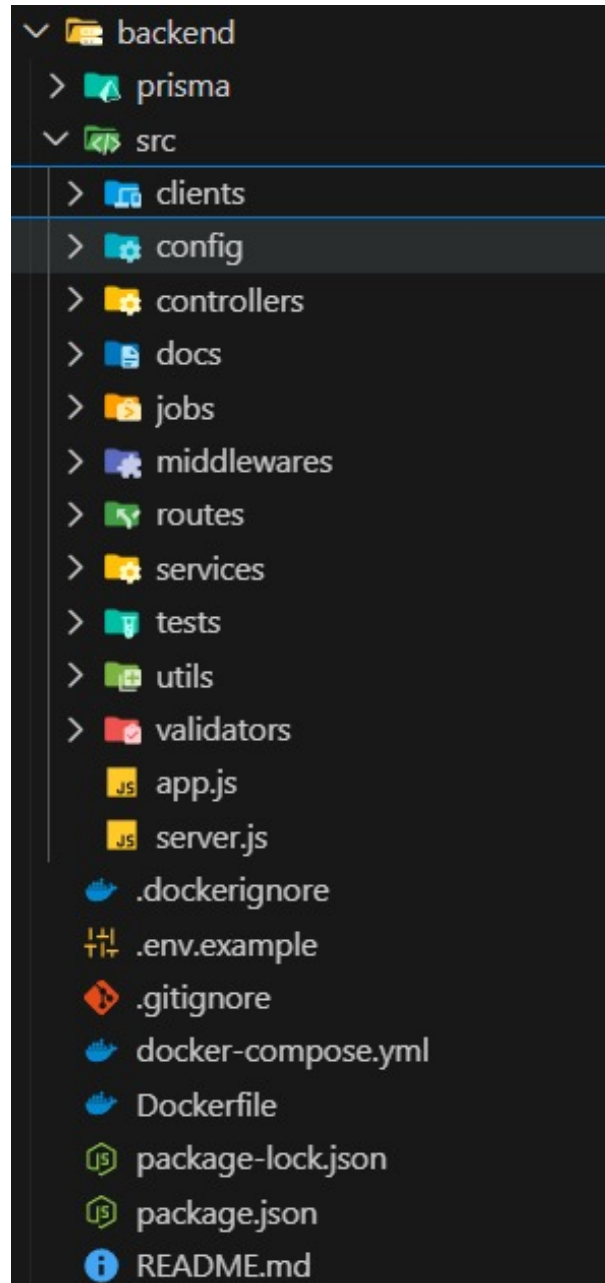
3.1 Arquitetura e tecnologias utilizadas

Para o desenvolvimento da solução, optou-se pela utilização de uma arquitetura moderna baseada em JavaScript e TypeScript, visando a coerência entre as camadas de *frontend* e *backend*, além de facilitar a tipagem estática e a manutenção do código.

O *backend* foi construído utilizando o ambiente Node.js com o framework Express. A escolha dessa tecnologia se deu pela sua eficiência na criação de APIs RESTful e pela grande disponibilidade de bibliotecas. Para a persistência de dados, utilizou-se o banco de dados relacional PostgreSQL, gerenciado pelo ORM (*Object-Relational Mapping*) Prisma. O Prisma foi escolhido por oferecer segurança de tipos (*type-safety*) e facilitar a modelagem das entidades do sistema, como *Projects*, *Sprints* e *Tasks*. Para garantir a consistência

entre os ambientes de desenvolvimento e produção, utilizou-se a plataforma **Docker**. A containerização da aplicação, incluindo o banco de dados e a API, permitiu isolar as dependências do projeto, facilitando a orquestração dos serviços e a implantação (*deploy*) da ferramenta. A estruturação do Back-End pode ser visualizada na imagem 2.

Figura 2 – Estrutura do Back-End da ferramenta



Fonte: Elaborada pelos autores (2025).

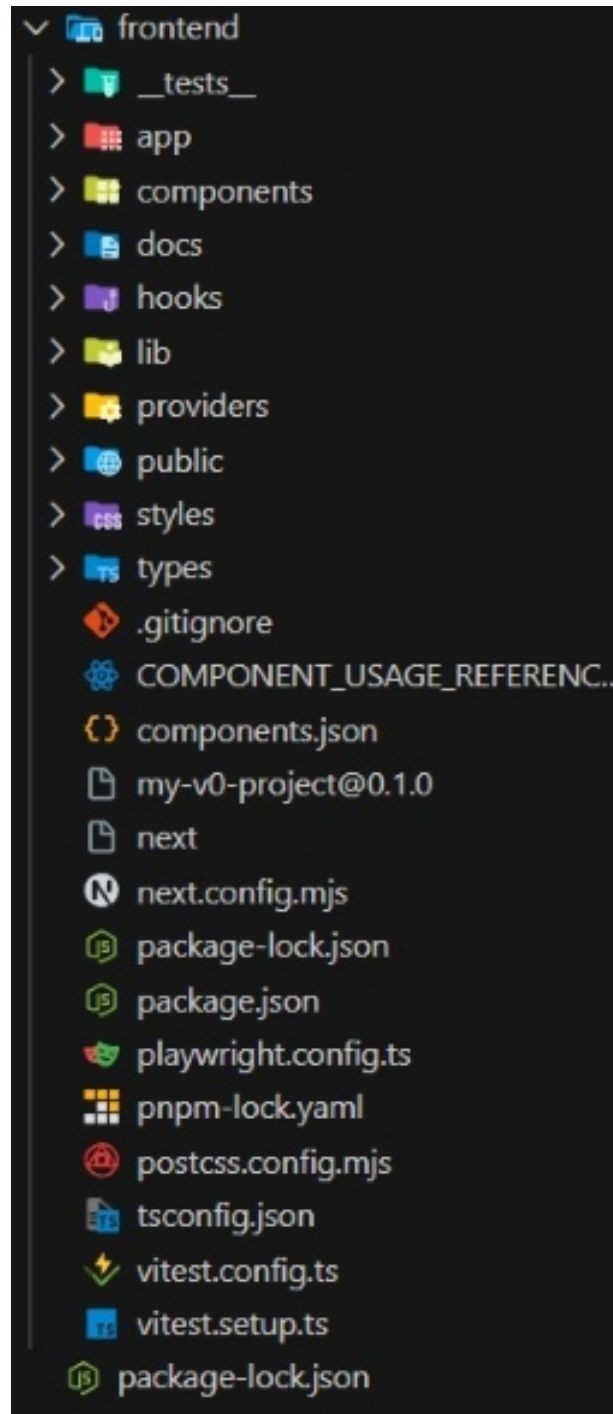
A arquitetura do servidor segue o padrão de camadas (*Layered Architecture*), garantindo a separação de responsabilidades:

- **Rotas:** Responsáveis por receber as requisições HTTP;

- **Controllers:** Gerenciam a lógica de entrada e saída de dados;
- **Services:** Encapsulam as regras de negócio complexas;
- **Repositórios (Prisma):** Realizam a comunicação direta com o banco de dados.

A estruturação do Front-End pode ser verificada através da figura 2.

Figura 3 – Estrutura do Front-End da ferramenta



Fonte: Elaborada pelos autores (2025).

Além disso, foram implementados *middlewares* transversais para garantir segurança e observabilidade, utilizando bibliotecas como *Helmet* (segurança de headers) e *CORS*.

3.2 Desenvolvimento do front-end e usabilidade

A interface do usuário foi desenvolvida utilizando o framework Next.js 14 com React 18. Essa escolha permite uma arquitetura otimizada para a web moderna, combinando performance de carregamento com a interatividade necessária de uma *Single Page Application* (SPA).

Para a construção visual, adotou-se o Tailwind CSS em conjunto com componentes acessíveis da biblioteca Radix UI (*shadcn/ui*), garantindo consistência visual e acessibilidade. A gestão de estado e cache das requisições foi implementada com o TanStack Query, otimizando a performance ao reduzir chamadas desnecessárias à API.

A usabilidade da ferramenta foca na experiência do usuário através de um layout tabulado que separa claramente as visões de *Dashboard*, *Kanban* e Relatórios. O quadro Kanban utiliza a biblioteca *react-beautiful-dnd* para permitir a funcionalidade de arrastar e soltar (*drag-and-drop*), essencial para a agilidade na movimentação de tarefas. O sistema conta ainda com suporte a cache de requisições, aumentando a resiliência da aplicação.

3.3 Integração com Inteligência Artificial

O diferencial da ferramenta reside na integração com um agente de Inteligência Artificial para a geração assistida de backlogs. Utilizou-se a API da Groq (*Chat Completions*), escolhida por sua baixa latência e alta capacidade de inferência (processo de gerar o *output* para o usuário).

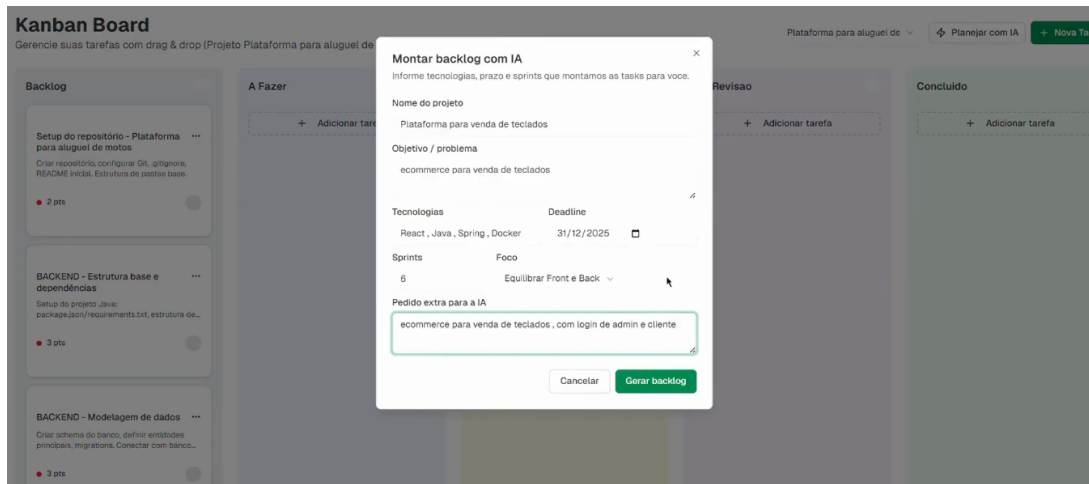
A implementação segue um padrão robusto com estratégias de *retry* (tentativas exponenciais) e mecanismos de *fallback* local caso a conexão com a IA falhe. Em termos de funcionalidades, a LLM foi projetada para atuar diretamente dentro da metodologia Scrum, operando com as seguintes capacidades:

- **Organização automatizada:** Criação e estruturação de tarefas a partir de comandos curtos ou descrições em linguagem natural fornecidas pelo usuário;
- **Priorização inteligente:** Sugestão de ordenação das tarefas, destacando os itens iniciais e de maior importância crítica para o projeto;
- **Classificação técnica:** Distinção automática de quais tarefas referem-se ao *Back-end* e quais referem-se ao *Front-end* do projeto;
- **Estimativa de Esforço:** Definição sugerida de Story Points (complexidade) para cada tarefa gerada, auxiliando na estimativa de esforço total da sprint;

- **Sugestão de Planejamento:** Identificação de quais tarefas são pré-requisitos lógicos, sugerindo uma ordem sequencial de execução ideal para o backlog.

A utilização da interface de criação de Backlog com o auxílio da LLM está representada graficamente na figura 4.

Figura 4 – Criação de Backlog com a IA



Fonte: Elaborada pelos autores (2025).

3.4 Estrutura de classes e serviços

O sistema foi estruturado em serviços especializados para garantir a manutenibilidade. Destacam-se as seguintes classes e responsabilidades:

- **ReportingService:** Responsável pelos cálculos analíticos, como métricas de total de pontos da sprint, acompanhamento de *burn-down* e progressão da sprint.
- **TeamMembersService:** Gerencia o ciclo de vida dos membros da equipe, incluindo validações de e-mail e atribuição de tarefas, facilitando a distribuição de responsabilidades.
- **Tratamento de Falhas:** Implementação de mecanismos de captura de erros em integrações externas, garantindo que falhas na API da IA não derrubem a aplicação.
- **Hierarchy of Errors:** Conjunto de tratamentos de erro personalizados que permite uma identificação precisa de falhas na comunicação com a IA.

3.5 Comparativo com ferramentas de Scrum existentes

A ferramenta desenvolvida possui semelhanças fundamentais com softwares existentes do mercado, como a utilização de colunas padrão de Kanban (*Backlog, To Do, In Progress, Done*) e a gestão de Sprints.

No entanto, diferencia-se por funcionalidades específicas voltadas para a automação e auditoria:

- **Geração de Backlog via IA:** Diferente da criação manual tradicional, a ferramenta propõe itens de trabalho estruturados com base apenas no contexto do projeto;
- **Rastreabilidade:** Registro de histórico de eventos (TaskHistory) no banco de dados para fins de controle e segurança;
- **Resiliência e Offline:** Suporte a cache de requisições, permitindo a visualização de dados previamente carregados mesmo sem conexão;
- **Monitoramento:** Implementação de logs estruturados e tratamento de erros centralizado para facilitar a identificação de falhas.

3.6 Resultados obtidos

Durante os testes de uso da ferramenta desenvolvida, observou-se que a interação com o agente de Inteligência Artificial trouxe um ganho significativo na etapa de planejamento. Ao especificar quais as tecnologias e os requisitos gerais do projeto, a IA detalhou e desagregou o escopo geral em subtarefas coerentes. O modelo conseguiu distinguir, por exemplo, a necessidade de criar tabelas no banco de dados antes de desenvolver as rotas da API, sugerindo uma ordem lógica de execução.

No que tange à gestão visual, o recurso de priorização demonstrou eficácia imediata. A aplicação automática de um código de cores — vermelho para alta prioridade, amarelo para média e verde para baixa — permitiu que, logo na primeira visualização do *board*, fosse possível identificar quais itens compõem o Produto Mínimo Viável (MVP) e requerem atenção imediata.

Além disso, a função de estimativa de esforço, representada pela atribuição de pontuação (*Story Points*) para cada tarefa, forneceu uma métrica inicial sólida para o planejamento das *Sprints*. Embora essas estimativas sejam sugestivas, elas serviram como um ponto de partida confiável, eliminando a "folha em branco".

3.7 Avaliação da ferramenta

A avaliação funcional da ferramenta desenvolvida revelou que a integração entre o *framework* Scrum e os Modelos de Linguagem de Grande Escala (LLM) cumpre o objetivo de reduzir a carga operacional do *Scrum Master* e do *Product Owner*. A resposta da API da Groq mostrou-se ágil o suficiente para manter a fluidez da experiência do usuário, não ocasionando em gargalos durante a geração do *backlog*.

É importante reafirmar que embora a ferramenta automatize a criação estrutural do projeto, a supervisão humana permaneceu necessária para refinar contextos muito específicos do negócio. No entanto, o esforço de "revisar e ajustar" o que a IA gerou provou-se consideravelmente menor do que o esforço de "criar do zero" e manualmente, validando a hipótese de que a ferramenta atua como um acelerador de produtividade, permitindo que a equipe foque sua energia cognitiva em resolver problemas complexos de engenharia, em vez de gastar um tempo excessivo na configuração manual de artefatos da gestão do projeto.

4 Trabalhos relacionados

Diversos estudos investigam como a Inteligência Artificial Generativa pode apoiar as práticas de scrum, seja na automação de tarefas, na geração de artefatos do projeto, ou na comunicação entre os integrantes da equipe. A seguir, são apresentados trabalhos relevantes que abordam ideias paralelas desta pesquisa.

4.1 Revisão dos estudos selecionados

Schröder (2023) aborda o conceito de "AutoScrum", estudando como modelos de linguagem podem ser utilizados para automatizar o planejamento de projetos e gerenciamento de tarefas. Em sua pesquisa, foram integradas LLMs juntamente a programas de orientação (*guidance programs*), no qual o modelo de linguagem recebe o estado atual do projeto criado e assim sugere requisitos, *user stories*, e funcionalidades para alcançar uma meta desejada.

Os objetivos centrais de Schröder (2023) dizem respeito a redução do tempo de planejamento, e a automação na geração de *backlogs*. Este autor demonstra que mesmo que a Inteligência Artificial possa gerar múltiplas versões de tarefas e prioridades, a validação humana é fundamental para que se garanta a coerência das decisões tomadas, unindo assim, a produtividade tecnológica com a expertise humana.

Em uma perspectiva mais ampla, Karnouskos (2024) explorou a relevância dos LLMs para a gestão de projetos como um todo. Foi avaliada a capacidade de modelos como GPT-3.5, GPT-4 e Gemini em lidar com desafios típicos da área, sujeitando-os a questões de exames de certificação PMI ¹ (*Project Management Institute*).

O estudo de Karnouskos focou em cenários de liderança, tomadas de decisões e monitoramento de progressos. Os resultados revelaram que embora o GPT-4 tenha obtido desempenho superior (cerca de 90% de acerto em testes teóricos), as IAs mostraram limitações em quesitos práticos que exigem julgamento social e inteligência emocional. O

¹ Project Management Institute: A principal organização mundial de gestão de projetos, responsável por certificações profissionais e pelo guia PMBOK (Project Management Institute, 2025).

trabalho conclui que os LLMs são competentes no que tange à teoria, mas não substituem as experiências interpessoais de um humano gerente de projetos.

Mais recentemente, Alliata, Arcuri e Beller (2025) investigaram discriminadamente a automação de tarefas no papel de Scrum Master. Esse trabalho testou o uso de LLMs para gerar relatórios de status a partir de gráficos de *burndown* e para criar *User Stories* baseadas em épicos (*Epics*). Segundo Cohn (2005), épicos são grandes corpos de trabalho que necessitam ser destrinchados em tarefas menores, enquanto gráficos de *burndown* são ferramentas visuais essenciais para monitorar a relação de trabalho restante e o tempo disponível na *sprint*.

A partir de modelos como GPT-4 e Claude 3, Alliata, Arcuri e Beller (2025) mediram a acurácia dos dados sintéticos gerados. Os resultados revelam que os LLMs podem desempenhar tarefas repetitivas, liberando o Scrum Master para atividades mais estratégicas, porém, reafirmam a necessidade da supervisão humana, especialmente em contextos complexos nos quais o modelo de linguagem pode falhar na interpretação de sutilezas do projeto.

4.2 Análise comparativa de abordagens

A análise dos três estudos evidencia o potencial transformador dos LLMs na metodologia de gestão ágil. Como visto, Schröder (2023) foca na etapa de **planejamento e definição** do escopo do projeto; Karnouskos (2024) avalia a **competência teórica e tomada de decisão** dos modelos de linguagem; e Alliata, Arcuri e Beller (2025) analisa a **automação operacional** da rotina do Scrum Master.

Apesar de utilizarem de métodos diferentes — simulação iterativa (SCHRÖDER, 2023), cenários de prova (KARNOUSKOS, 2024), e casos de usos experimentais (ALLIATA; ARCURI; BELLER, 2025) — todos estes convergem para um resultado em comum: o aumento perceptível da produtividade e eficiência, entretanto, se fazendo criticamente necessária a supervisão humana. As limitações apontadas, tais como a dificuldade em lidar com contextos práticos não textuais, e a necessidade de refinamento dos outputs, abrem lacunas para novas ferramentas que integrem melhor essas capacidades.

4.3 Relação com o trabalho desenvolvido

Os estudos investigados embasam diretamente as funcionalidades implementadas na ferramenta proposta neste TCC. O enfoque de automação de planejamento de Schröder (2023) valida a função de **criação assistida de backlogs** implementada. A perspectiva de Karnouskos (2024) sobre o suporte a decisões são representados nas funcionalidades de **priorização inteligente** reunidos no sistema. Já a visão de Alliata, Arcuri e Beller (2025) de automatizar o dia-a-dia do Scrum Master se relaciona com a centralização de **relatórios** e **estruturação de sprints** da ferramenta desenvolvida. Diferentemente dos

trabalhos analisados, este por sua vez busca aplicar estes conceitos em uma aplicação prática e centralizada, tendo como objetivo o aprimoramento da ação da equipe ágil.

5 Considerações finais

A gestão de projetos de software, especialmente sob a ótica das metodologias ágeis, enfrenta o desafio constante de equilibrar a velocidade de entrega com a qualidade do planejamento. Este trabalho buscou investigar e implementar uma solução tecnológica que integrasse Modelos de Linguagem de Grande Escala (LLM) ao framework Scrum, visando mitigar a sobrecarga operacional de papéis de liderança, como o *Scrum Master* e o *Product Owner*.

Ao longo do desenvolvimento, a integração entre a arquitetura moderna de software e as capacidades cognitivas da Inteligência Artificial demonstrou ser não apenas viável, mas eficaz na modernização de rituais tradicionais do desenvolvimento ágil.

5.1 Resultados obtidos

Os resultados obtidos durante o desenvolvimento e a avaliação da ferramenta mostram que a integração de Large Language Models (LLMs) ao framework Scrum é eficaz para apoiar as equipes e reduzir a carga operacional dos papéis de liderança. Nos testes realizados, a ferramenta demonstrou capacidade de contribuir de forma consistente para as principais etapas do processo ágil.

A ferramenta demonstrou capacidade de auxiliar na otimização do planejamento das sprints ao transformar escopos gerais em subtarefas estruturadas, claras, priorizadas e bem organizadas. Esse processo facilitou a compreensão do backlog e ajudou a equipe a visualizar com mais precisão quais atividades deveriam ser executadas.

Outro resultado relevante foi a mitigação do esforço demandado do Product Owner e do Scrum Master, pois a IA automatizou tarefas repetitivas, como reorganização do backlog e revisão de descrições de tarefas. Dessa forma, os líderes puderam dedicar mais tempo a decisões estratégicas e ao alinhamento com a equipe.

Além disso, a ferramenta apresentou boa capacidade de adaptação ao contexto, respondendo adequadamente a mudanças de escopo e a novas instruções durante a sprint. Esse comportamento reforça a viabilidade de utilizar LLMs como facilitadores no apoio a equipes que trabalham com Scrum.

Em resumo, os resultados sugerem que a solução proposta tem potencial para aumentar a eficiência operacional, melhorar a organização dos artefatos do Scrum e contribuir diretamente para a gestão do desenvolvimento ágil de software.

5.2 Limitações do estudo

Apesar dos avanços alcançados com o desenvolvimento da ferramenta, algumas limitações foram identificadas ao longo do estudo. Observou-se que algumas gerações automáticas requerem revisões manuais para se adequarem ao contexto específico e às regras de negócio do projeto. Isso demonstra que, embora o sistema seja útil na automação de tarefas, a ferramenta não elimina totalmente a necessidade de supervisão e intervenção humana.

Outra limitação refere-se à natureza da validação, que foi restrita a testes funcionais conduzidos internamente pela própria equipe de desenvolvimento. A ausência de testes com usuários externos ou em cenários reais de produção restringe a capacidade de avaliar a curva de aprendizado e a usabilidade da ferramenta por profissionais que não participaram de sua concepção, o que pode mascarar dificuldades de uso que seriam evidentes para novos usuários.

Além disso, a avaliação realizada foi predominantemente qualitativa e técnica, baseada na análise das respostas produzidas pelo modelo. Não foi conduzido um estudo de campo longitudinal que permitisse mensurar, em termos quantitativos — como horas economizadas ou redução de esforço operacional — o impacto da ferramenta em comparação com métodos tradicionais de documentação e planejamento. Essa ausência de métricas limita a capacidade de estimar, de forma concreta, a melhoria de produtividade proporcionada pelo sistema.

5.3 Sugestões para trabalhos futuros

Considerando as limitações identificadas neste estudo, bem como a necessidade de aprofundar as análises quantitativas e ampliar a validação da solução, algumas direções são recomendadas para pesquisas futuras. Inicialmente, sugere-se a realização de sessões de testes com usuários reais, com o objetivo de coletar feedback qualitativo sobre a experiência de uso e verificar se a interface é intuitiva para profissionais que não participaram do desenvolvimento da ferramenta. Essa etapa permitirá identificar oportunidades de aprimoramento relacionadas à usabilidade e ao engajamento dos usuários.

Além disso, recomenda-se a condução de estudos quantitativos e longitudinais que possibilitem mensurar indicadores específicos de agilidade, tais como o tempo total do processo (Lead Time), o tempo de execução das tarefas (Cycle Time) e o tempo efetivamente economizado no fluxo de trabalho. Esses estudos devem comparar o desempenho obtido pelos métodos tradicionais, sem a utilização da ferramenta, com os resultados alcançados mediante sua adoção, permitindo avaliar, de forma estatística e objetiva, o impacto real do projeto na produtividade das equipes.

Outra oportunidade de melhoria está na implementação de um mecanismo de "Engenharia de Prompt Assistida" (ou engenharia reversa de comandos). Neste cenário, caso o

usuário insira um comando ambíguo ou mal formulado, a IA seria capaz de identificar a insuficiência de informações e solicitar que o usuário refine sua entrada. Isso garantiria uma geração de backlog mais precisa e evitaria alucinações ou resultados genéricos por parte do modelo.

Por fim, destaca-se a importância de ampliar os cenários de avaliação, aplicando a ferramenta em ambientes organizacionais de maior escala, envolvendo múltiplas equipes e projetos com diferentes níveis de complexidade. Essa expansão possibilitará analisar a robustez da solução, sua capacidade de adaptação a contextos variados e seu potencial de generalização para além de um ambiente controlado.

Agradecimentos

O desenvolvimento deste trabalho ocorreu com o apoio de outras pessoas envolvidas além dos integrantes da equipe, como o professor Elton Figueiredo, ao qual gostaríamos de expressar nossa gratificação por esses meses de orientação e parceria.

Gostaríamos de agradecer a Deus pela capacidade e discernimento; aos nossos familiares por todo o apoio e prestígio na defesa do TCC; aos nossos colegas que também compareceram na apresentação; aos avaliadores da banca Flávio Dusse e Pedro Nascimento, por terem aceitado de bom grado o convite para a banca e pelas sinalizações de melhoria no artigo; e ao coordenador Osvaldo Requião pelo apoio durante a defesa de nosso trabalho.

Referências

- ALLIATA, Z.; ARCURI, A.; BELLER, M. The ai scrum master: Using large language models (llms) to automate agile project management tasks. *Munique*, 2025. Acesso em: 29 set. 2025. Automação de tarefas operacionais (como relatórios e requisitos) para liberar o profissional (humano) para tarefas mais estratégicas. Disponível em: <https://www.researchgate.net/publication/387904405_The_AI_Scrum_Master_Using_Large_Language_Models_LLMs_to_Automate_Agile_Project_Management_Tasks>.
- Atlassian. *O que é Agile?* 2025. Acesso em: 2025. Disponível em: <<https://www.atlassian.com/br/agile>>.
- BECK, K. et al. *Manifesto para o desenvolvimento ágil de software*. 2001. Acesso em: 3 out. 2025. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>.
- BENDER, E. M. et al. On the dangers of stochastic parrots: Can language models be too big? In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. [S.l.: s.n.], 2021. (FAccT '21), p. 610–623.
- BOMMASANI, R. et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. Acesso em: 3 out. 2025. Disponível em: <<https://arxiv.org/pdf/2108.07258>>.
- CALISKAN, A.; BRYSON, J. J.; NARAYANAN, A. Semantics derived automatically from language corpora contain human-like biases. *Science*, American Association for the Advancement of Science, v. 356, n. 6334, p. 183–186, 2017.
- CINKUSZ, A.; CHUDZIAK, K.; NIEWIADOMSKA-SZYNKIEWICZ, E. Large language models in multi-agent systems for agile software engineering. *Information Systems*, 2025.
- COHN, M. *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice Hall, 2005. Obra referência para definições de estimativa ágil, épicos e gráficos de monitoramento.
- COHN, M. *Succeeding with Agile: Software Development Using Scrum*. Boston: Addison-Wesley, 2010.
- DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. [S.l.: s.n.], 2019. p. 4171–4186.
- GARCÍA, C.; LEE, D. Identifying and mitigating impediments in agile scrum projects. *Journal of Systems and Software*, v. 200, p. 111666, 2023. Acesso em: 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0740624X23000588>>.
- HIGHSMITH, J. *Agile Software Development Ecosystems*. Boston: Addison-Wesley, 2002.
- KARNOUSKOS, S. The relevance of large language models for project management. In: IEEE. *IEEE International Conference on Project Management*. Walldorf, 2024. Acesso em: 24 set. 2025. Disponível em: <<https://www.researchgate.net/>>

publication/381354721_The_Relevance_of_Large_Language_Models_for_Project_Management>.

MAYOR, A. *Gods and Robots: Myths, Machines, and Ancient Dreams of Technology*. Princeton: Princeton University Press, 2018. A obra detalha especificamente os autômatos criados por Hefesto, como Talos e as servas douradas, como precursores da IA. Disponível em: <<https://assets.press.princeton.edu/chapters/i14162.pdf>>.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, v. 5, n. 4, p. 115–133, 1943. Disponível em: <<https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf>>.

PRASETYO, Y. J. et al. Problems in the adoption of agile-scrum software development process in small organization: A systematic literature review. *ResearchGate*, 2024. Acesso em: 2025. Disponível em: <https://www.researchgate.net/publication/377518159_Problems_in_The_Adoption_of_Agile-Scrum_Software_Development_Process_in_Small_Organization_A_Systematic_Literature_Review>.

Project Management Institute. *Who We Are*. 2025. Acesso em: 05 dez. 2025. Disponível em: <<https://www.pmi.org/about>>.

RADFORD, A. et al. *Improving Language Understanding by Generative Pre-Training*. 2018. Acesso em: 3 out. 2025. Disponível em: <https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf>.

RIBAS, T. *O que é Scrum?* 2025. Blog Thomaz Ribas. Acesso em: 09 dez. 2025. Disponível em: <<https://thomazribas.com/blog/scrum>>.

SCHRÖDER, M. Autoscrum: Automating project planning using large language models. *arXiv preprint arXiv:2306.03197*, 2023. Acesso em: 22 set. 2025. Disponível em: <<https://arxiv.org/pdf/2306.03197>>.

SCHWABER, K.; SUTHERLAND, J. Scrum development process. *OOPSLA Business Object Design and Implementation Workshop*, v. 20, n. 2, p. 77–86, 1995. Disponível em: <<https://jeffsutherland.com/oopsla/schwapub.pdf>>.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. 2020. Acesso em: 3 out. 2025. Disponível em: <<https://scrumguides.org/scrum-guide.html>>.

STRUBELL, E.; GANESH, A.; MCCALLUM, A. Energy and policy considerations for deep learning in nlp. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. Florence: [s.n.], 2019.

TAKEUCHI, H.; NONAKA, I. The new new product development game. *Harvard business review*, v. 64, n. 1, p. 137–146, 1986. Disponível em: <<https://damiantgordon.com/Methodologies/Papers/The%20New%20Product%20Development%20Game.pdf>>.

TOUVRON, H. et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. Acesso em: 3 out. 2025. Disponível em: <<https://arxiv.org/pdf/2302.13971>>.

TURING, A. M. Computing machinery and intelligence. *Mind*, v. 59, n. 236, p. 433–460, 1950. Disponível em: <<https://courses.cs.umbc.edu/471/papers/turing.pdf>>.

VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30, p. 5998–6008. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.